**Employing temporal model for group activity**

By Mostafa Saad Ibrahim

Last update: July 21, 2015

[Big Picture](#h.vjdcshp278es)

[The Temporal Model](#h.zes09rv0h2xt)

[Input processing](#h.74ygw7r2gxto)

[Future Prediction](#h.6f97rujfkxkt)

[Architecture variations](#h.6ygucr9mecr)

[Datasets](#h.d330ww96ukjz)

[Action Items](#h.7400y0qof4vk)

[Mostafa Todos](#h.nbig8vuqqe0e)

[Question Marks!](#h.8dkba760dbvb)

[Performed Experiments](#h.4ma4c0ee3lef)

[Dataset 1 (Scenes Dataset)](#h.bz4vvvmpj961)

[Scene Classification - No Tracklets - No LSTM](#h.yg83ngk233j8)

[Scene Classification - Tracklets - LSTM layer 1](#h.gxhd4i9qa8sn)

[Dataset 2 (Players actions/poses Dataset)](#h.5izq9olfxoh5)

[Player Classification - LSTM](#h.h56cfzlok5ev)

[Scene Classification - 2 layer LSTM - setup 1](#h.al53j7ruvd8l)

# Big Picture

* **Input**: a volleyball match
* **Outputs** may be
  + Scene classification (e.g. serve action, winpoint for a team)
    - 6 labels pers team
      * Server, set, pass, block, spike, winpoint
    - As we have 2 teams (left - right) => total 12 labels
  + Players classification
    - Actions
      * E.g. Walk. **Todo** others.
      * [Arash] Suggestions: Jump, Block, Spike, Drill, Set, Serve, Walk.
    - Roles
      * **Todo**
  + Regression - Future Prediction
    - Next position to hit the ball
    - [Arash] It can be converted to a classification problem, if you have a set of detections and it is assumed that the next position to hit is one of the detected players.
  + More on Activity forecasting? **Todo**
* **Approach**: Employing temporal model for group activity

# The Temporal Model

* A multi-layer LSTM that aggregates information in time domain and optimize with respect to the different tasks (classifications / regression)
* Assume a **2 layer LSTMs**
  + Architecture justification:
    - *The target outputs (e.g. scene labels) require multiple sources of information*
    - *Social roles and group activities incorporate information about the behaviours of multiple people in the scene*
    - *[Arash] The temporal dynamic of activities/actions can be represented using LSTM layers.*
  + First layer is 1 LSTM per player action (or may be role)
    - E.g. 12 LSTMs - 1 LSTM per a player
    - Probably the parameters in this layer will be shared to avoid memory/overfitting issues
    - Output of each LSTM is a real-valued vector.
      * **Todo** see Question Marks section
  + Second layer aggregates inputs from the first layer LSTMs
    - We may do max pooling over all feature vectors from the LSTMs vectors
      * More suitable for scene classification
      * Concern: Gradient update
    - We may do some sort of spatial max pooling. My thoughts
      * E.g. divide frame to 4 blocks and do spatial pooling for players in each region?
      * Sort the 12 players by top left point, max pool each 3 players. Then we have 4 feature vectors, expressing 2 teams
* Temporal window: Probably in range 10-20 frames block

# Input processing

* For every frame or bounding box, we should use a pre-trained model from Caffe Zoo to extract features
* We may also fine tune the pre-trained model against volleyball dataset to do some customization for feature vectors
* Using Caffe-Feature Extraction tool
  + This way we split processing on 2 levels
  + First: frames to feature vectors (of length 4096) and save to disk
  + Second: load feature vectors and learn the temporal model

# Future Prediction

* ToDo

# Architecture variations

* The proposed temporal model could be changed to different styles. Experimentally we could know the good one
* Examples ToDo

# Datasets

**Our Volleyball dataset**

* So far around 800 frames annotated through 4 videos
  + Annotated like 1.5 hours from 1 video and 7 minutes from other 3 videos
  + Only frames of scene action (e.g. spike) are labeled
  + 2 “Ball positions” are labeled between 2 of above annotations
    - Current hit position - i.e. a player doing an action such pass
    - Next hit position - i.e. to whom the player is going
  + Ignore high motion labels (e.g. serve). Annotate but don’t use

**Other suitable dataset**

# Action Items

**Target LSTM Library**

* Make decision which library to use
* [CURRENT](http://sourceforge.net/projects/currennt/) library doesn’t seem to support the multi-outputs actions in experiment 2 below. Need to verify.
* [CAFFE-LSTM](https://github.com/junhyukoh/caffe-lstm) library input example is so simple. I posted issue on github asking about feeding our input format

**Players Trajectory**

* A challenging part that affects overall performance
* Gkioxari and J. Malik [action tube](http://www.cs.berkeley.edu/~gkioxari/ActionTubes/action_tubes.pdf) paper used RCNN and proposed a simple tube for each moving person. They fails when multi-person do crossings
  + They explained how to filter much of RCNN windows based on motion for faster processing. We will need this trick.
* We may detect using RCNN cross multiframes like action tube paper
* Or detect once and track using a tracker through the video. We may need to investigate the available trackers from a [recent benchmark](https://sites.google.com/site/trackerbenchmark/benchmarks/v10)
* [Arash] We may need to consider fine-tuning the RCNN person detector on our volleyball dataset to improve its detection performance.
* [Arash] We should not spend too much time on finding the best detection/tracking system. We can probably have a noisy system and hope that our 2 layer LSTM will fix some of the problems. For example instead of 12 detection we can extract more detection (18-24) and hope that our model will ignore the wrong detections.

**Dataset Annotations**

* We need to decide how to do so
  + We may work manually => not so effective
  + We may hire some students to label
  + We may hire Amazon Mechanical Turk => not so smooth
  + [Arash] Amazon Mechanical Turk may be easier than you think and even cheaper than hiring students.
* What actions/role to label?
* Should we generate player's trajectory and then label them?
* Or annotations should be for boxes / labels together?
* We need to develop/find a program to do the annotations. We may need to check first the needs of Amazon truck (e.g. may need a tool as web interface)

[Srikanth]

* We have completed annotating 173 frames for beginning our experiment, with 173\*12 player annotations.
* We had initially planned to annotate all the players and referees, and later backtracked to annotate only the players.
* We should consider about the additional annotations from our side if needed.

**Experiment 1**

* Pure frame to scene label classification through LSTM
* Major also, let the frame be for 16 players. Generate tracklet.
* We will use caffe to extract features for **whole** frame
* Feed the features/labels to 1 LSTM network

**Experiment 2**

* Given 12 players trajectory cross around 10-20 frames
  + Extract feature vector for each player
  + Concatenate the 12 vectors
  + Learn multi-outputs actions for the concatenated actions (e.g multi-loss)
    - e.g. learn 12 outputs vector (walk, walk, run, ...etc)
* **Dependency**
  + Players Trajectories
  + Dataset annotations
  + Library that supports multi-outputs actions
* [Arash] Another alternative can be given 12 player trajectories on 10-20 frame:
  + Extract feature vectors for each player.
  + Put 1 LSTM on top of each player.
  + Predict action labels for each player independently (without multi-loss)

**Experiment 3**

* This is our 2 layers LSTM training
* First part should be doing experiment 2 (the lower part of architecture)
* Second part training upper part (top LSTM) with fixing lower level weights
* Concerns
  + We may need to do kind of network expansion code to expand experiment 2 network to the 2 layers LSTM structure?!
  + [Arash] We may be able to use tricks like reshaping data in a batch in Caffe to feed the output of the first LSTM over all detections to the second LSTM.

**Sampling vs whole video**

* We may pick specific frame, build track around it and judge it
* Or we may use the frames between 2 labels and consider them background/none class [Arash suggestion]
* This way we do continue labeling
* In both...we can either recognize or forecast

# Mostafa Todos

* For June, I will work on
  + Experiment 1
  + Players Trajectory
  + Be familiar with an LSTM library code
* On parallel, **we** need to decide how to annotate data for further experiments

SRIKANTH TODO:

* Implement two layer LSTM using Jeff Donahue’s library
* Discuss different methods on Tracking/Detection

# Question Marks!

* We once in the meeting said the first layer LSTMs will generate a real valued feature vector (or arbitrary length)
* In experiment 2 we want to learn a 12 outputs per the concatenated input vector?
* But I understand that, experiment 2 is part of experiment 3.
* Any hints where i am understanding wrong?
* [Arash] Check my comment for the Experiment 2. I think my suggestion can help you move to the experiment 3 easily from experiment 2.
  + [Mostafa]: Perfect

# 

# Performed Experiments

## Dataset 1 (Scenes Dataset)

* We have 10 classes
  + **Spike, Block, Dig, Pass, Winpoint** x 2 teams
  + We **removed** 2 serve classes to avoid much motion
* 4 videos
  + 667 frames in total (train = 465, val = 127, test = 75)

## Scene Classification - No Tracklets - No LSTM

|  |  |
| --- | --- |
| * Work on dataset 1 * No LSTM. No tracklets. * Given an image, we learn corresponding class out of 10 classes   + Caffe input: (3 channels image, image label) | |
| 00014353.jpg  Output => left\_spike scene class | |
| * Uses Alex network. Initialize it with pre-trained network * Validation **Accuracy**: 55% * Test **Accuracy**: * Hard disk path: /cs/vml2/msibrahi/workspaces/caffe-lstm/examples/volleyball\_w0\_c3\_fullframe | |

## Scene Classification - Tracklets - LSTM layer 1

|  |  |
| --- | --- |
| * Work on dataset 1 * Temporal window = 9 frames (tried longer too) * Given an image, we learn corresponding class out of 10 classes   + Caffe input: 9 tracklet frames each is (3 channels image, image label) | |
| 00007170.jpg   * We push in the database for training, 9 tracklet images as above image. Each image has 16 players detected. | |
| * Uses Alex network. Initialize it with pre-trained network * Add 1 LSTM layer with **250 nodes** * Validation **Accuracy**: 51.5% * Test **Accuracy**: * Hard disk path: /cs/vml2/msibrahi/workspaces/caffe-lstm/examples/volleyball\_w4\_c3\_tracklets\_lstm\_finetune | |

## Dataset 2 (Players actions/poses Dataset)

* We have 9 classes
  + Representing how a player looks like
  + **Jumping, Waiting, Standing, Moving, Diving, Setting, Digging, Blocking, Spiking**
* 4 videos
  + 2150 frames in total (train = 1373, val = 364, test = 413)

## Player Classification - LSTM

|  |  |
| --- | --- |
| * Work on dataset 2 * LSTM over tracklets of length 5 * Given a cropped image around a player, we learn corresponding class out of 10 classes   + Caffe input: (3 channels image of a player pose, image label) | |
| player.jpeg  Output => Waiting class | |
| * Uses Alex network. Initialize it with pre-trained network * Validation **Accuracy**: 65% * Test **Accuracy**: * Hard disk path: /cs/vml2/msibrahi/workspaces/caffe-lstm/examples/players\_w2\_c3\_lstm\_finetune | |

## Scene Classification - 2 layer LSTM - setup 1

|  |  |
| --- | --- |
| * Work on dataset 1 * LSTM over tracklets of length 9 * No end-to-end training.   + Prepare inputs from LSTM 1 and save to disk   + LSTM 2 load the data and train them * Dataset preparation   + Given a video frame, Detect 16 players   + Find tracklet of each player through 9 frames   + Using Player Classification **LSTM-1**, describe each player box     - Then we have 16 \* 9 feature vectors, each of length 250     - Save to database 9 aggregated feature vectors, each of length 4000 (16 \* 250). In other words, construct aggregated feature vector per image.       * Each feature vector label is the video frame label * Now database in format (aggregated feature vector, scene label (10 classes) )   + Originally we have 667 frames in total (train = 465, val = 127, test = 75)     - Multiply by factor 9     - (train = 4185, val = 1143, test = 675) | |
| Input => feature vector of length 4000  Output => Winpoint class | |
| * Network   + 1 LSTM layer of 1000 nodes   + 1 Inner Product network to our 10 classes * Validation **Accuracy**: 43% * Test **Accuracy**: * Hard disk path: /cs/vml2/msibrahi/workspaces/caffe-lstm/examples/players\_w2\_c3\_lstm\_finetune | |